SQL

STEP BY STEP

BASIC  ADVANCE

Make Sure To **Download** Or Clone This Repo

https://github.com/yaswanthteja/Complete_SQL/

Complete SQL With Notes

1. Introduction to SQL-What Is SQL & Database
2. Data Types, Primary-Foreign Keys & Constraints
   a. Install postgresql and pgadmin4
3. Create Table In SQL & Create Database
4. INSERT UPDATE, DELETE & ALTER Table
5. SELECT Statement & WHERE Clause with Example
6. How To Import Excel File (CSV) to SQL
7. Functions in SQL & String Function
8. Aggregate Functions – Types & Syntax
9. Group By and Having Clause
10. Time Stamp and Extract Function, Date Time Function
11. SQL JOINS – Types & Syntax
12. SELF JOIN, UNION & UNION ALL
13. Subquery
14. Window Function – Types & Syntax
15. Case Statement/Expression with examples
16. CTE- Common Table Expression with examples

Filtering & Sorting

Creating a Products table for practice

```
-- Create the Products table
CREATE TABLE Products (
  ProductID INT PRIMARY KEY,
  ProductName VARCHAR(50) NOT NULL,
  Category VARCHAR(30),
  Price INT,
  Stock INT
);

-- Insert  data
INSERT INTO Products (ProductID, ProductName, Category, Price, Stock)
VALUES
(1, 'Laptop', 'Electronics', 55000, 25),
(2, 'Headphones', 'Electronics', 1500, 100),
(3, 'Coffee Mug', 'Kitchen', 300, 200),
(4, 'Office Chair', 'Furniture', 7000, 15),
(5, 'Smartphone', 'Electronics', 25000, 40),
(6, 'Blender', 'Kitchen', 1800, 60),
(7, 'Desk Lamp', 'Furniture', 1200, 80),
(8, 'Monitor', 'Electronics', 12000, 30);
```

SELECT Statement

The SELECT statement is used to select data from a database.

- **Syntax**

```
SELECT column_name FROM table_name;
```

To select all the fields available in the table

- **Syntax**

```
SELECT * FROM table_name;
```

To select distinct/unique fields available in the table

- **Syntax**

```
SELECT DISTINCT Column_name FROM table_name;
```

Operators In SQL

The SQL reserved words and characters are called operators, which are used with a WHERE clause in a SQL query

Most used operators:

1. **Arithmetic operators :** arithmetic operations on numeric values
   Example: Addition (+), Subtraction (-), Multiplication (*), Division (/), Modulus (%)
2. **Comparison operators:** compare two different data of SQL table
   - Example: Equal (=), Not Equal (!=), Greater Than (>), Greater Than Equals to (>=)
3. **Logical operators:** perform the Boolean operations
   - Example: AND, OR, NOT, ANY
4. **Bitwise operators:** perform the bit operations on the Integer values
   - Example: Bitwise AND (&), Bitwise OR(|)

SQL Operators

Arithmetic Operators

+, -, *, /, %

Comparison Operators

=, !=, <>, >, <, >=, <=

Logical Operators

AND, OR, NOT

Bitwise operator

&, ‘|, ~,^

Compound Operator

+=, -=, *=, /=, %=

Special Operators

BETWEEN ... AND ..., IN(...), LIKE, IS NULL, EXISTS

WHERE Clause

The WHERE clause is used to filter records.

It is used to extract only those records that fulfill a specified condition

- **Syntax**

```
SELECT column_name FROM table_name
WHERE conditions;
```

- **Example**

```
select * from products
where category='Electronics';
```

LIMIT Clause

The LIMIT clause is used to set an upper limit on the number of Rows returned by SQL.

Example: below code will return 5 rows of data

```
SELECT column_name FROM table_name
LIMIT 5;
```

OFFSET Clause

The **OFFSET clause** in SQL is used to **skip a specific number of rows** before starting to return results.
It's often used with **LIMIT** to implement **pagination** — like showing page 2 of search results.

Syntax:

```
SELECT columns
FROM table
ORDER BY column
OFFSET n ROWS
LIMIT m;
```

ORDER BY Clause

The ORDER BY is used to sort the result-set in ascending (ASC) or descending order (DESC).

Example: below code will sort the output data by column name in ascending order

```
SELECT column_name FROM table_name
ORDER BY column_name e ASC;
```

Example: Sort products by price (lowest to highest) and Skip the first 2 cheapest products to Show the next 3

```
SELECT ProductName, Price
FROM Products
ORDER BY Price ASC
OFFSET 2     -- skips first 2 products
LIMIT 3;     -- gives next 3 products
```